Source and Channel Adaptive Rate Control for Multicast Layered Video Transmission Based on a Clustering Algorithm

Jérôme Viéron, Thierry Turletti, Kavé Salamatian, Christine Guillemot

EURASIP Journal on Applied Signal Processing 2004:2, 158–175

Source and Channel Adaptive Rate Control for Multicast Layered Video Transmission Based on a Clustering Algorithm

Jérôme Viéron
Thomson multimedia R&D, 1 avenue Bellefontaine - CS 17616, 35576 Cesson-Sévigné, France
Email: jerome.vieron@inria.fr

Thierry Turletti
INRIA, 2004 route des Lucioles - BP 93, 06902 Sophia Antipolis Cedex, France
Email: thierry.turletti@inria.fr

Kavé Salamatian
Laboratoire d'Informatique de Paris 6 (LIP6), 8 rue du Capitaine Scott, 75015 Paris, France
Email: kave.salamatian@inria.fr

Christine Guillemot
INRIA, Campus de Beaulieu, 35042 Rennes Cedex, France
Email: christine.guillemot@inria.fr

Received 24 October 2002; Revised 8 July 2003

This paper introduces source-channel adaptive rate control (SARC), a new congestion control algorithm for layered video transmission in large multicast groups. In order to solve the well-known feedback implosion problem in large multicast groups, we first present a mechanism for filtering RTCP receiver reports sent from receivers to the whole session. The proposed filtering mechanism provides a classification of receivers according to a predefined similarity measure. An end-to-end source and FEC rate control based on this distributed feedback aggregation mechanism coupled with a video layered coding system is then described. The number of layers, their rate, and their levels of protection are adapted dynamically to aggregated feedbacks. The algorithms have been validated with the NS2 network simulator.

Keywords and phrases: multicast, congestion control, layered video, aggregation, FGS.

1. INTRODUCTION

Transmission of multimedia flows over multicast channels is confronted with the receivers heterogeneity problem. In a multicast topology (multicast delivery tree in the $1 \rightarrow N$ case, acyclic graph in the $M \rightarrow N$ case), network conditions such as loss rate (LR) and queueing delays are not homogeneous in the general case. Rather, there may be local congestions affecting downstream delivery of the video stream in some branches of the topology. Hence, the different receivers are connected to the source via paths with varying delays, loss, and bandwidth characteristics. Due to this potential heterogeneity, dynamic adaptation of multimedia flows over multicast channels, for optimized quality-of-service (QoS) of multimedia sessions, faces challenging problems. The adaptation of source and transmission parameters to the network state often relies on the usage of feedback mechanisms. However, the use of feedback schemes in large multicast trees faces the potential problem of feedback implosion. This paper introduces source-channel adaptive rate control (SARC), a new congestion control algorithm for layered video transmission in large multicast groups. The first issue addressed here is therefore the problem of aggregating heterogeneous reports into a consistent view of the communication state. The second issue concerns the design of a source rate control mechanism that would allow a receiver to receive the source signal with a quality commensurate with the bandwidth and loss capacity of the path leading to it.

Layered transmission has been proposed to cope with receivers heterogeneity [1, 2, 3]. In this approach, the source is represented using a base layer (BL) and several successive enhancement layers (EL) refining the quality of the source reconstruction. Each layer is transmitted over a separate multicast group, and receivers decide the number of groups to join (or leave) according to the quality of their reception. At the other side, the sender can decide the optimal number of layers and the encoding rate of each layer according to the feedback sent by all receivers. A variety of multicast schemes making use of layered coding for audio and video communication have been proposed, some of which rely on a multicast feedback scheme [3, 4]. Despite rate adaptation to the network state, applications have to face the remaining packet losses. Error control schemes using *forward error correction* (FEC) strongly reduce the impact of packet losses [5, 6, 7]. In these schemes, redundant information are sent along with the original information so that the lost data (or at least part of it) can be recovered from the redundant information. Clearly, sending redundancy increases the probability of recovering the lost packets, but it also increases the bandwidth requirements, and thus the LR of the multimedia stream. Therefore, it is essential to couple the FEC scheme to the rate control scheme in order to jointly determine the transmission parameters (redundancy level, source coding rate, type of FEC scheme, etc.) as a function of the state of the multicast channel, to achieve the best subjective quality at receivers. For such adaptive mechanisms, it is important to have simple channel models that can be estimated in an online manner.

The sender, in order to adapt the transmission parameters to the network state, does not need reports of each receiver in the multicast group. It rather needs a partition of the receivers into homogeneous classes. Each layer of the source can then be adapted to the characteristics of one class or of a group of classes. Each class represents a group of homogeneous receivers according to discriminative variables related to the received signal quality. The clustering mechanism used here follows the above principles. A classification of receiver reports (RRs) is performed by aggregation agents (AAs) organized into a hierarchy of local regions. The approach assumes the presence of AAs at strategic positions within the network. The AAs classify receivers according to similar reception behaviors and filter correspondingly the (real-time transport control protocol) RTCP RRs. By classifying receivers, this mechanism solves the feedback implosion problem and at the same time provides the sender with a compressed representation of the receivers.

In the experiments reported in this paper, we consider two pairs of discriminative variables in the clustering process: the first one constituted of the LR and the *goodput* and the second constituted of the LR and the throughput of a conformant TCP (transport control protocol) connection under similar loss and round-trip time (RTT) conditions. We show approaches in which receivers rate requests are only based on the goodput measure risk leading to a severe subutilization of the network resources. To use a TCP throughput model, receivers have to estimate their RTT to the source first. In order to do so, we use the algorithm described in [4] jointly with a new application-defined RTCP packet, called *probe RTT*.

This distributed feedback aggregation mechanism is coupled with a video fine-grain scalable (FGS) layered coding system to adapt dynamically the number of layers, the rate of each layer, and its level of protection. Notice that the aggregation mechanism that has to be supported by the network nodes remains generic and can be used for any type of media. The optimization is performed by the sender and takes into account both the network aggregated state as well as the rate-distortion characteristics of the source. The latter allows to optimize the quality perceived by each receiver in the multicast tree.

The remainder of this paper is organized as follows. Section 2 provides an overview of related research on multicast rate and congestion control. Section 3 sets the main lines of SARC, our new hybrid sender/receiver driven rate control based on a clustering algorithm. The protocol functions to be supported by the receivers and the receiver clustering mechanism governing the feedback aggregation are described, respectively, in Sections 4 and 5. Section 6 describes the multilayer source and channel rate control and the multi-layered MPEG-4 FGS source encoder [8, 9] that have been used in the experiments. Finally, experimental results obtained with the NS2 network simulator with various discriminative clustering variables (goodput, TCP-compatible throughput), including the additional usage of FEC are discussed in Section 7.

2. RELATED WORK

Related work in this area focuses on error, rate, and congestion control in multicast for multimedia applications. Layered coding is often proposed as a solution for rate control in video multicast applications over the Internet. Several approaches—sender-driven [10], receiver-driven [11, 12], or hybrid schemes [3, 13, 14]—have been proposed to address the problem of rate control in a multicast transmission. Receiver-driven approaches consist in multicasting different layers of video using different multicast addresses and let the receivers decide which multicast group(s) to subscribe to. RLM (receiver-driven layered multicast) [11] and RLC (radio link control) [12] are two well-known receiver-driven layered multicast congestion control protocols. However, they both suffer from pathological behaviors such as transient periods of congestion, instability, and periodic losses. These problems mainly come from the bandwidth inference mechanism used [15]. For example, RLM uses *join experiments* that can create additional traffic congestion during transition periods corresponding to the latency for pruning a branch of the multicast tree. RLC [12] is a TCP-compatible version of RLM, based on the generation of periodic bursts that are used for bandwidth inference on synchronization points indicating when a receiver can join a layer. Both the synchronization points and the periodic bursts can lead to periodic

congestion and periodic losses [15]. PLM (Packet-pair layered multicast) [16] is a more recent layered multicast congestion control protocol, based on the generation of packet pairs to infer the available bandwidth. PLM does not suffer from the same pathological behaviors as RLM and RLC but requires a fair queuing network.

Bhattacharya et al. [17] present a general framework for the analysis of additive increase multiplicative decrease (AIMD) multicast congestion control protocols. This paper shows that because of the so-called "path loss multiplicity problem," unclever use of congestion information sent by receivers to 1 sender may lead to severe degradation and lack of fairness. This paper formalizes the multicast congestion control mechanism in two components: the loss indication filter (LIF) and the rate adjustement algorithm. Our paper presents an implementation that minimises the loss multiplicity problem by using an LIF which is implemented by a clustering mechanism (Section 5.2) and a rate adjustement algorithm following the algorithm described in Sections 4 and 6.

TFMCC [18] is an equation-based multicast congestion control mechanism that extends the TCP-friendly TFRC [19] protocol from the unicast to the multicast domain. TFMCC uses a scalable RTT measurement and a feedback suppression mechanism. However, since it is a single-rate congestion control scheme, it cannot handle heterogeneous receivers and adapts its sending rate to the current limiting receiver.

FLID-DL [20] is a multirate congestion control algorithm for layered multicast sessions. It mitigates the negative impact of long Internet group management protocol (IGMP) leave latencies and eliminates the need for probe intervals used in RLC. However, the amount of IGMP and PIM-SM (protocol independent multicast-sparse mode) control traffic generated by each receiver is prohibitive. WEBRC [21] is a new equation-based rate control algorithm that has been recently proposed. It solves the main drawbacks of FLID-DL using an innovative way to transmit data in waves. However, WEBRC, such as FLID-DL, is intended for reliable download applications and possibly streaming applications but cannot be used to transmit real-time hierarchical flows such as H.263+ or MPEG-4.

A source adaptive multilayered multicast (SAMM) algorithm based on feedback packets containing information on the estimated bandwidth (EB) available on the path from the source is described in [3]. Feedback mergers are assumed to be deployed in the network nodes to avoid feedback implosion. A mechanism based on *partial suppression* of feedbacks is proposed in [4]. This approach avoids the deployment of aggregation mechanisms in the network nodes, but on the other hand, the partial feedback suppression will likely induce a flat distribution of the requested rates.

MLDA [13] is a TCP-compatible congestion control scheme in which, as in the scheme we propose, senders can adjust their transmission rate according to feedback information generated by receivers. However, MLDA does not provide a way to adapt the FEC rate in the different layers according to the packet loss observed at receivers. Since the feedback only includes TCP-compatible rates, MLDA does not need feedback aggregation mechanisms and uses exponentially distributed timers and a partial suppression mechanism to prevent feedback implosion. However, when the receivers are very heterogeneous, the number of requested rates (in the worst case on a continuous scale) can potentially lead to a feedback implosion. Moreover, the partial suppression algorithm does not allow quantifying the number of receivers requesting a given rate in order to estimate how representative this rate is.

In [14], a rate-based congestion and loss control mechanism for multicast layered video transmission is described. The strategy relies on a mechanism that aggregates feedback information in the networks nodes. However, in contrast with SAMM, the optimization is not performed in the nodes. Source and channel FEC rates in the different layers are chosen among a set of requested rates in order to maximize the overall peak signal-to-noise ratio (PSNR) seen by all the receivers. Receivers are classified according to their available bandwidth, and for each class of rate, two types of information are delivered to the sender: the number of receivers represented by this class and an average LR computed over all those receivers. It is supposed here that receivers with similar bandwidths have similar LRs, which may not always be the case. In this paper, we solve this problem using a distributed clustering mechanism.

Clustering approaches have been already considered separately in [22, 23]. In [22], a centralized classification approach based on k-means clustering is applied on a quality of reception parameter. This quality of reception parameter is derived, based on the feedback of receivers consisting of reports including the available bandwidth and packet loss. The main difference, compared with our approach, is that in our case, the classification is made in a distributed fashion. Hence, receivers with similar bandwidths but with different LRs are not classified within the same class. Therefore, with more accurate clusters, a better adaptation of the error control process at the source level is possible. The global optimization performed is different and leads to improved performances. Moreover, [22] uses the RTCP filtering mechanism proposed in the RTP (real-time transport protocol) standard, that is, they adapt the RTCP sending rate according to the number of receivers. However, when the number of receivers is large, it is not possible to get a precise snapshot of quality observed by receivers.

3. PROTOCOL OVERVIEW

This section gives an overview of the SARC protocol proposed in this paper. Its design relies on a feedback tree structure, where the receivers are organized into a tree hierarchy, and internal nodes aggregate feedbacks.

At the beginning of the session, the sender announces the range of rates (i.e., a rate interval $[R_{\min}, R_{\max}]$) estimated from the average rate-distortion characteristics of the source. The value $R_{\min}$ corresponds to the bit rate under which the

received quality would not be acceptable, whereas $R_{\max}$ corresponds to the rate above, under which there is no significant improvement of the visual quality. This information is transmitted to the receivers at the start of the session. The interval $[R_{\min}, R_{\max}]$ is then divided into subintervals in order to only allow relevant values for layers rates. This quantization avoids having nonquality discriminative layers.

After this initialization, the multicast layered rate control process can start. The latter assumes that the time is divided into feedback rounds. A feedback round comprises four major steps.

(i) At the beginning of each round, the source announces the number of layers and their respective rates via RTCP sender reports (SRs). Each source layer is transmitted to an Internet protocol (IP) multicast group.

(ii) Each receiver measures network parameters and estimates the bandwidth available on the path leading to it. The EB and the layer rates will trigger subscriptions or unsubscriptions to/from the layers. EB and LRs are then conveyed to the sender via RTCP RR.

(iii) AAs placed at strategic positions within the network classify receivers according to similar reception behaviors, that is, according to a measure of distance between the feedback parameter values. On the basis of this clustering, these agents proceed with the aggregation of the feedback parameters, providing a representation of homogeneous clusters.

(iv) The source then proceeds with a dynamic adaptation of the number of layers and of their rates in order to maximize the quality perceived by the different clusters.

Sections 4, 5, and 6 describe in details each of the four steps.

4. PROTOCOL FUNCTIONS SUPPORTED BY THE RECEIVER

Two bandwidth estimation strategies have been considered: the first approach measures the goodput of the path and the second estimates the TCP-compatible bandwidth under similar conditions of LRs and delays. This section describes the functions supported by the receiver in order to measure the corresponding parameters and the multicast groups join and leave policy that has been retained. The bandwidth values estimated by the receivers are then conveyed to the sender via RTCP RRs augmented with dedicated fields.

4.1. Goodput-based estimation

A notion of goodput has been exploited in the SAMM algorithm described in [3]. Assuming the priority-based differentiated services for the different layers, the goodput is defined as the cumulated rate of the layers received without any loss. If a layer has suffered from losses, it will not be considered in the goodput estimation. The drawback of such a measure is that the EB will be highly dependent on the sending rates, hence it does not allow an accurate estimation of the link capacity. When no loss occurs, in order to best approach the link capacity, SAMM considers values higher than the goodput measured. Nevertheless, a LR of 0% is not realistic on the Internet. Experiments have shown that this notion of goodput in a best-effort network, in presence of cross traffic, leads to EBs decreasing towards zero during the sessions. Here, the goodput is defined instead as the rate received by the end system. A simple mechanism has been designed to try to approach the bottleneck rate of the link. If the LR is under a given threshold T_{loss}, the bandwidth value B_t estimated at time t is incremented as

$$B_t = B_{t-1} + \Delta, \tag{1}$$

where Δ represents a rate increment and B_{t-1} represents the last estimated value. Let g_t be the observed goodput value at time t. Thus, when the LR becomes higher than the threshold T_{loss}, B_t is set to g_t.

In the experiments we have taken $t_{\text{loss}} = 3\%$ and the Δ parameter increases similarly to the TCP increase, that is, of one packet per RTT.

4.2. TCP-compatible bandwidth estimation

The second strategy considered for estimating the bandwidth available on the path relies on the analytical model of TCP throughput [24], known also as the TCP-compatible rate control equation. Notice, however, that the application of the model in a multicast environment is not straightforward.

4.2.1. TCP throughput model

The average throughput of a TCP connection under given delay and loss conditions is given by [24]:

$$T = \frac{\text{MSS}}{\text{RTT}\sqrt{2p/3} + T_o \min\left(1, 3\sqrt{3p/8}\right) p\left(1 + 32p^2\right)}, \tag{2}$$

where p, RTT, MSS, and T_o represent, respectively, the congestion event rate [19], the round-trip time, the maximum segment size (i.e., maximum packet size), and the retransmit time out value of the TCP algorithm.

4.2.2. Parameters estimation

In order to be able to use the above analytical model, each receiver must estimate the RTT on its path. This is done using a new application-defined RTCP packet that we called probe RTT. To prevent feedback implosion, only leaf aggregators are allowed to send probe RTT packets to the source. In case receivers are not located in the same LAN of their leaf aggregator, they should add the RTT to their aggregator; this can be easily estimated locally and without generating undesirable extra traffic. The source periodically multicasts RTCP reports including the RTT computed (in milliseconds) for the latest probe RTT packets received along with the corresponding SSRCs. Then, each receiver can update its RTT estimation using the result sent for its leaf aggregator. The estimation of

the congestion event rate p is done as in [25] and the parameter MSS is set to 1000 bytes.

4.2.3. Singular receivers

In highly heterogeneous environments, under constraints of bounded numbers of clusters, the rate received by some end systems may strongly differ from their requests, hence from the TCP-compatible throughput value. The resulting excessively low values of congestion event rates lead in turn to overestimated bandwidth values, hence to unstability. In order to overcome this difficulty, the TCP-compatible throughput B_t at time t is estimated as

$$B_t = \min\left(T, \max\left(S_{\text{rate}} + T_{\text{rate}}, B_{t-1}\right)\right), \tag{3}$$

where S_{rate} is the rate subscribed to, T_{rate} is a threshold chosen so that the increase between two requests is limited (i.e., $T_{\text{rate}} = K \times \text{MSS}/\,\text{RTT}$ with K a constant), and B_{t-1} is the last estimated value of the TCP-compatible throughput. When the estimated throughput value T is not reliable, the history used in the estimation of LRs is reinitialized using the method described in [19]. We will see in the experimentation results that the above algorithm is still reactive and responsive to changes in network conditions.

4.2.4. Slow-start mechanism

The slow-start mechanism adopted here differs from the approaches described in [18, 19]. At the beginning of the session or when a new receiver joins the multicast transmission tree, the requested rate is set to $R_{\min}$. Then, after having a first estimation of RTT and p, T can be computed and the resulting requested rate B_t^{slow} is given by

$$B_t^{\text{slow}} = \max\left(T, g_t + K \times \frac{\text{MSS}}{\text{RTT}}\right), \tag{4}$$

where g_t is the observed goodput value at time t and K is the same constant as the one used in Section 4.2.3. The estimation given by (4) is used until we observe the first loss. After the first loss, the loss history is reinitialized taking g_t as the available bandwidth and proceeding with (3).

4.3. Join/leave policy

Each receiver estimates its available bandwidth B_t and joins or leaves layers accordingly. However, the leaving mechanism has to take into account the delay between the instant in which a feedback is sent and the instant in which the sender adapts the layer rates accordingly. Undesirable oscillations of subscription may occur if receivers decide to unsubscribe a layer as soon as the TCP-compatible throughput estimated is lower than the current rate subscribed to. It is essential to leave enough time for the source to adapt its sending rates, and only then decide to drop a layer if the request has not been satisfied. That is why in order to be still reactive, we have chosen a delay of $K \times$ RTT before leaving a layer except in the case where the LR becomes higher than a chosen acceptable bound T_{loss} (K is the same constant as the one used in Section 4.2.3). These coupled mechanisms permit avoiding a waste of bandwidth due to IGMP traffic.

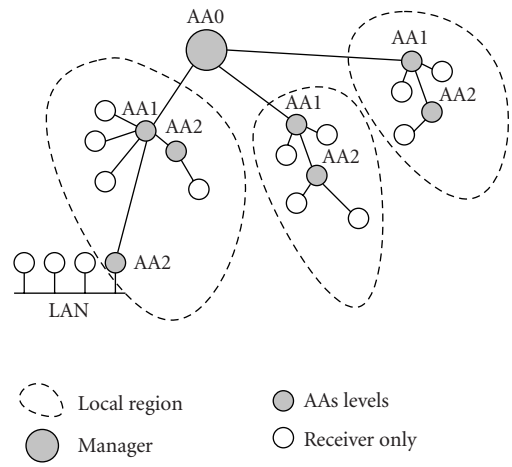


Figure 1: Multilevel hierarchy of aggregators.

4.4. Signalling protocol

The aggregated feedback information (i.e., EB and LR) are periodically conveyed towards the sender in RTCP RRs, using the RTCP report extension mechanism. The RRs are augmented with the following fields:

(i) EB: a 16-bit field which gives the value of the estimated bandwidth expressed in Kbps;

(ii) LR: a 16-bit field which gives the value of the real loss rate;

(iii) NB: a 16-bit field which gives the number of clients requesting this rate (i.e., EB). This value is set to one by the receiver.

5. AGGREGATED FEEDBACK USING DISTRIBUTED CLUSTERING

Multicast transmission has been reported to exhibit strong spatial correlations [26]. A classification algorithm can take advantage of this spatial correlation to cluster similar reception behaviors into homogeneous classes. In this way, the amount of feedback required to figure out the state of receivers can be significantly reduced. This will also help in bypassing loss path multiplicity problem explained in [17] by filtering out the receivers' report of losses. In our scheme, receivers are grouped into a hierarchy of local regions (see Figure 1). Each region contains an aggregator that receives feedback, performs some aggregation statistics, and send them in point-to-point to the higher level aggregator (*merger*). The root of the aggregator tree hierarchy (called the *manager*) is based at the sender and receives the overall aggregated reports.

This architecture has a slight modification compared to the generic RTP architecture. Similar to the PIM-SM context, RRs are not sent in multicast to the whole session, but are sent in point-to-point to a higher level aggregator. As these RTCP feedbacks are local to an aggregator region and will not cross the overall multicast tree, they may be set to be more frequent without breaking the 5% of the overall traffic constraint specified by the RTP standard.

5.1. Aggregators organization within the network

AAs must be set up at strategic positions within the network in order to minimize the bandwidth overhead of RTCP RRs. Several approaches have been proposed to organize receivers in a multicast session to make scalable reliable multicast protocols [27]. We have chosen a multilevel hierarchical approach such as that described in the RMTP [28] protocol in which receivers are also grouped into a hierarchy of local regions. However, in our approach, there are no designated receivers: all receivers send their feedback to their associated aggregator.

The root of the aggregator tree hierarchy (called the manager) is based at the sender and receives the overall summary reports. The maximal allowed height of the hierarchical tree is set to 3 as recommended in [29]. In our approach, the overall summary report is a classification containing the number of receivers in each class and the mean behaviour of the class. The mechanism of aggregation is described in Section 5.2.

In our experiments, aggregators are manually set up within the network. However, if extra router functionalities are available, several approaches can be used to automatically launch aggregators within the network. For example, we can implement the *aggregator* function using a *custom concast* [30]. *Concast* addresses are the opposite of multicast addresses, that is, they represent groups of senders instead of groups of receivers. So, a concast datagram contains a multicast group source address and a unicast destination address. With such a scheme, all receivers send their RRs feedback packets using the RTCP source group address to the sender's unicast address, and only one aggregated packet is delivered to the sender. The custom concast signaling interface allows the application to provide the network with the description of the merging algorithm function.

5.2. Clustering mechanism

The clustering mechanism is aimed towards taking advantage of the spatial and temporal correlation between the receiver's state of reception. Spatial correlation means that there is redundancy between reception behavior of neighbor receivers. This redundancy can be removed by compression methods. This largely reduces the amount of data required for representing feedback data sent by receivers. The compression is achieved by clustering similar (by a predefined similarity measure) reception behaviors into homogeneous classes. In this case, the clustering can be viewed as a vector quantization [31] that constructs a compact representation of the receivers as a classification of receivers issuing similar RRs. Moreover, for sender-based multicast regulation, only a classification of receivers is sufficient to apply adaptation decisions.

The clustering mechanism can also take advantage of time redundancy. For this purpose, classification of receivers should integrate the recent history of receivers as well as the actual RRs. Different reception states experienced by receivers during past periods are treated as reports of different and heterogeneous receivers. By this way, temporal variation of the quality of a receiver reception are integrated in the classification. A receiver that observes temporal variation may change its class during time.

In a stationary context, the classification would converge to a stable distribution. This stationary distribution will be a function of the spatial as well as the temporal dependencies. However, since over large time scales, the stationary hypothesis cannot be always validated, a procedure should be added to track variation of the multicast channel and adapt the classification to it. This procedure can follow a classical exponential weighting that drive the clustering mechanism to forget about far past-time reports. In this weighting mechanism, the weight of clusters is multiplied by a factor ($\gamma < 1$) at the end of each reporting round, and clusters with weight below a threshold are removed.

Before describing the classification algorithm, several concepts should be introduced. First, we should choose the discriminative characteristic and the similarity (or dissimilarity) measure needed to detect similar reception behavior.

5.2.1. Discriminative network characteristics

In the system presented in this paper, we have considered two pairs of discriminative variables: the first one constituted of the LR and the goodput (cf. Section 4.1) and the second constituted of the LR and a TCP-compatible bandwidth share measure (cf. Section 4.2). Both LR and bandwidth characteristics (goodput or TCP-compatible) are clearly relevant not only as network characteristics but also as video quality parameters.

5.2.2. Similarity measure

Two kinds of measures should be defined: the similarity measure between two observed reports x and y ($d(x, y)$) and between an observed report x and a cluster C ($d(x, C)$). The former similarity measure can stand for the simple L^p distance ($d(x, y) = \sqrt[p]{\sum_i (x_i - y_i)^p}$) or any other more sophisticated distance suitable to a particular application. The retained similarity measure used in this work is given by $d(x, y) = \max_i (abs(x_i - y_i)/dt_i)$, where dt_i is a chosen threshold for the dimension i. The latter similarity measure is more difficult to apprehend. The simplest way is to choose in each cluster a representative $\hat{x}_C$ and to assign the distance $d(x, \hat{x}_C)$ to the distance between the point and the cluster ($d(x, C) = d(x, \hat{x}_C)$). We can also define the distance to cluster as the distance to the nearest or the furthest point of the cluster ($d(x, C) = \min_{y \in C} d(x, y)$ or

$d(x, C) = \max_{y \in C} d(x, y)$). The distance can also be a likelihood derived over a model mixture approach. The type of measure used will impact over the shape of the cluster and over the classification.

5.2.3. Classification algorithm

Each cluster is represented by a representative point and a weight. The representative point can be seen as a vector, the components of which are given by the discriminative variables considered in the clustering process.

The clustering algorithm is initialized with a maximal number of classes ($N_{\max}$) and a cluster creation threshold (d_{th}). AAs regularly receive RTCP reports from receivers and/or other AAs in their coverage area as described in Section 5.1. To classify the RRs in the different clusters, we use a very simple nearest neighbor (NN) k-means clustering algorithm (see pseudocode shown in Algorithm 1). Even if this algorithm might be subject to largely reported deficiencies as false clustering, dependencies on the order of presentation of samples, and nonoptimality which has lead researchers to develop more complex clustering mechanism as mixture modelling, we believe that this rather simple algorithm attain the goal of our approach which is to filter out RRs to a compact classification in a distributed, asynchronous way. A new report joins the cluster that has the lowest Euclidean (L^2) distance to it and updates the cluster representative by a weighted average of the points in the cluster. When a new point joins a cluster, it changes slightly the representative point which is defined as the cluster center and updates the weight of the cluster; afterwards, the point is dropped to achieve compression. If this minimal distance is more than a predefined threshold, a new cluster is created. This bounds the size of the cluster. We also use a maximal number of clusters (or classes) which is fixed to 5, as it is not realistic to have more layers in such a layered multicast scheme.

At the end of each reporting round, the resulting classification is sent back to the higher level AA (i.e., the manager) in the form of a vector of clusters representatives and of their associated weights, and clusters are reset to a null weight. Clusters received by different lower level AAs are classified following a similar clustering algorithm which will aggregate representative points of clusters, that is, cluster center, with the given weight. This amounts to applying the NN clustering algorithm to the representative points reported in the new coming RR.

At the higher level of the aggregators hierarchy, the clustering generated by aggregating lower level aggregator reports is renewed at the beginning of each reporting round.

As explained before, the classification of receivers should also integrate the recent history of receivers. This memory is introduced into the clustering process by using the cluster obtained during the past reporting round as an a priori in the highest level of the aggregator hierarchy.

Nevertheless, since, over large time scales, the stationary hypothesis cannot be always validated, a procedure must be added to ensure that we forget about far past-time reports and not to bias the cluster representative by out-of-date reports. This is handled by an exponential weighting heuristic: at each reporting round, the weight of a cluster is reduced by a constant factor (see Algorithm 2). If the weight of a cluster falls below a cluster suppression threshold level, the cluster is removed.

```
Search for the nearest cluster d(r, Ĉ) = min_C d(r, C)
if (d(r, Ĉ) ≥ d_th)
    if (Number of existing cluster < N_max)
        Add a new cluster C_new and set Ĉ = C_new
Recalculate the representative of cluster Ĉ,
x̂_Ĉ = (weight(Ĉ) x̂_Ĉ + r) / (weight(Ĉ) + 1)
Increment the weight of cluster Ĉ
d_th = predefined threshold
N_max = maximal number of clusters (5)
r = received receiver report
```

Algorithm 1: NN clustering algorithm.

```
At the beginning of each reporting round
    for all clusters C
        % Weight the current normalized cluster by γ
        weight(C) = weight(C) * γ
        if weight(C) < w_min
            Remove cluster C
    Aggregate new normalized reports
    Send aggregate reports to the sender
w_min = predefined cluster suppression threshold
γ = memory weight
```

Algorithm 2: Aggregation algorithm at the highest level with memory weighting.

5.2.4. Cluster management

The clustering algorithm implements three mechanisms to manage the number of clusters: a cluster addition, a cluster removal, and a cluster merge mechanisms. The cluster addition and the cluster removal mechanisms have been described before. The cluster merging mechanism aims at reducing the number of clusters by combining two clusters that have been driven very close to each other. The idea behind this mechanism is that clusters should fill up uniformly the space of possible reception behaviors. The cluster merging mechanism merges two clusters that have a distance lower than a quarter of the cluster creation threshold (d_{th}). The distance between the two clusters is defined as the weighted distance of the cluster representatives. The merging threshold is chosen based on the heuristic that (1) d_{th} defines the fair diameter of a cluster and (2) two clusters that are distant by $d_{\text{th}}/4$ may be created by merging a cluster of diameter smaller than d_{th}. The cluster merging mechanism replaces the two clusters with a new cluster

represented by a weighted average of the two cluster representatives and a weight corresponding to the sum of the two clusters.

The combination of these three mechanisms of cluster management creates a very dynamic and reactive representation of the reception behaviour observed during the multicast session.

6. LAYERED SOURCE CODING RATE CONTROL

The feedback channel created by the clustering mechanism offers periodically to the sender information about the network state. More precisely, this mechanism delivers a LR, a bandwidth limit, and the number of receivers within a given cluster. This information is in turn exploited to optimize the number of source layers, the coding mode, the rate, and the level of protection of each layer. This section first describes the media and FEC rate control algorithm that takes into account both the network state and the source rate-distortion characteristics. The FGS video source encoding system used and the structure of the streaming server considered are then described.

6.1. Media and FEC rate-distortion optimization

We consider, in addition, the usage of FEC. In the context of transmission on the Internet, error detection is generally provided by the lower layer protocols. Therefore, the upper layers have to deal mainly with erasures or missing packets. The exact position of missing data being known, a good correction capacity can be obtained by systematic maximal distances separable (MDS) codes [32]. An (n, k) MDS code takes k data packets and produces $n - k$ redundant data packets. The MDS property allows to recover up to $n - k$ losses in a group of n packets. The effective loss probability $P_{\text{eff}}(k)$ of an MDS code, after channel decoding, is given by

$$P_{\text{eff}}(k) = P_e\left(\sum_{j=0}^{k-1}\binom{n-1}{j}P_e^{n-1-j}(1-P_e)^j\right), \tag{5}$$

where P_e is the average loss probability on the channel. One question to be solved is then, given the effective loss probability, how to split in an optimal way the available bandwidth for each layer between raw and redundant data. This amounts to finding the level of protection (or the code parameter k/n) for each layer.

The rates for both raw data and FEC (or equivalently, the parameter k/n) are optimized jointly as follows. For a maximum number of layers L supported by the source, the number of layers, their rate, and their level of protection are chosen in order to maximize the overall PSNR seen by all the receivers. Note that the rates are chosen in the set of N requested rates (feedback information). This can be expressed as

$$(\Omega_1, \ldots, \Omega_l) = \arg\max_{(\Omega_1,\ldots,\Omega_l)} G, \tag{6}$$

where $\Omega_i = (r_i, \kappa_i/n)$, $i = 1, \ldots, l$, with r_i representing the cumulated source and channel rate and κ_i/n the level of protection for each layer i. The quality measure G to be maximized is defined as

$$G = \sum_{j=1}^{N}\left(\sum_{i=1}^{l}\text{PSNR}(\Omega_i)\cdot P_{j,i}\right)\cdot C_j, \tag{7}$$

where

$$l = \arg\max_{k\in[1,\ldots,L]}\left\{\sum_{i=1}^{k} r_i \le R_j\right\}. \tag{8}$$

The terms R_j and C_j represent, respectively, the requested rate and the number of receivers in the cluster j. The term $\text{PSNR}(\Omega_i)$ denotes the PSNR increase associated with the reception of the layer i. Note that the PSNR corresponding to a given layer i depends on the lower layers. The term $P_{j,i}$ denotes the probability, for receivers of cluster j, that the i layers are correctly decoded and can be expressed as

$$P_{j,i} = \prod_{k=1}^{i}\left(1 - \tilde{p}_{\text{eff}_{j,k}}\left(\frac{\kappa_k}{n}\right)\right), \tag{9}$$

where $\tilde{p}_{\text{eff}_{j,k}}$ is the effective loss probability observed by all the receivers of the cluster j receiving the k considered layers. The values $\text{PSNR}(\Omega_i)$ are obtained by estimating the rate-distortion $D(R)$ performances of the source encoder on a training set of sequences. The model can then be refined on a given sequence during the encoding process, if the coding is performed in real time, or stored on the server in the case of streaming applications.

The upper complexity bound, in the case of an exhaustive search, is given by $L!/N!(N - L)!$, where L is the maximum number of layers and N the number of clusters. However, this complexity can be significantly reduced by first sorting the rates R_j requested by the different clusters. Once the rates R_j have been sorted, the constraint given by (8) allows to limit the search space of the possible combinations of rate r_i per layer. Hence, the complexity of an exhaustive search within the resulting set of possible values remains tractable. For large values of L and N, the complexity can be further reduced by using dynamic programming algorithm [33].

Notice that here we have not considered the use of hierarchical FEC. The FEC used here (i.e., MDS codes) are applied on each layered separately. Only their rates k_i/n are optimized jointly. The algorithm could be extended by using layered FEC as described in [34].

6.2. Fine-grain scalable source

The layers are generated by an MPEG-4 FGS video encoder [8, 9]. FGS has been introduced in order to cope with the adaptation of source rates to varying network bandwidths in the case of streaming applications with pre-encoded streams.

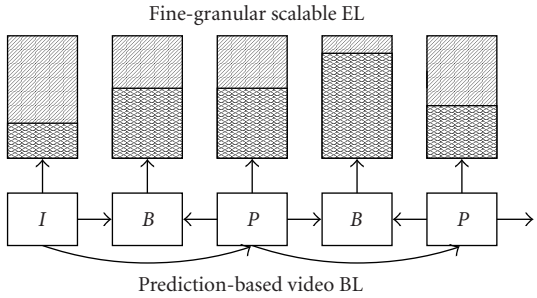


Figure 2: FGS video coding scalable structure.

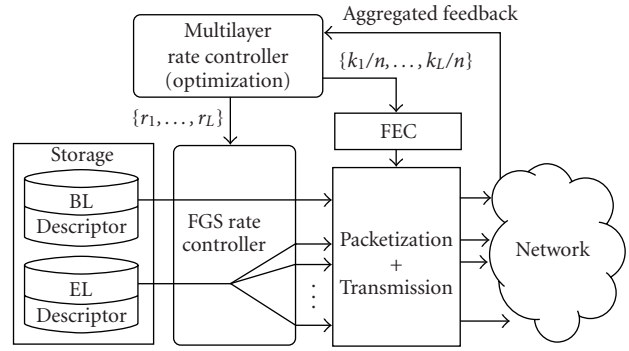


Figure 3: Multicast FGS streaming server.

Indeed, even if classical scalable (i.e., SNR, spatial, and temporal) coding schemes provide elements of response to the problem of rate adaptation to network bandwidth, those approaches suffer from limitations in terms of adaptation granularity. The structure of the FGS method is depicted in Figure 2. The BL is encoded at a rate denoted by R_{BL}, using a hybrid approach based on a motion compensated temporal prediction followed by a DCT-based compression scheme. The EL is encoded in a progressive manner up to a maximum bit rate denoted by R_{EL}. The resulting bitstream is progressive and can be truncated at any points, at the time of transmission, in order to meet varying bandwidth requirements. The truncation is governed by the rate-distortion optimization described above, considering the rate-distortion characteristics of the source. The encoder compresses the content using any desired range of bandwidths $[R_{\min} = R_{\mathrm{BL}}, R_{\max}]$. Therefore, the same compressed streams can be used for both unicast and multicast applications.

6.3. Multicast FGS streaming server

The experiments reported in this paper are done assuming an FGS streaming server. Figure 3 shows the internal structure of the multicast streaming system considered including the layered rate controller and the FEC module. For each video sequence prestored on the server, we have two separate bitstreams (i.e., one for BL and one for EL) coupled with its respective descriptors. These descriptors contain various information about the structure of the streams. Hence, it contains the offset (in bytes) of the beginning of each frame within the bitstream of a given layer. The descriptor of the BL contains also the offset of the beginning of a slice (or video packet) of an image. The composition timestamp (CTS) of each frame used as the presentation time at the decoder side is also contained in the descriptor.

Upon receiving a new list $(r_0, r_1, \ldots, r_L)$ of rate constraints, the FGS rate controller computes a new bit budget per frame (for each expected layer) taking into account the frame rate of the video source. Then, at the time of transmission, the FGS rate controller partitions the FGS enhancement into a corresponding number of "sublayers." Each layer is then sent to a different IP multicast group. Notice that, regardless of the number of FGS ELs that the client subscribes to, the decoder has to decode only one EL (i.e., the sublayers of the EL merge at the decoder side).

6.4. Rate control signalling

In addition to the value of the RTT computed for the probe RTT packets, the RTCP SRs periodically sent include information about the sent layers, that is, their number, their rate, and their level of protection, according to the following syntax:

(i) NL: an 8-bit field which gives the number of enhancement layers;
(ii) BL: a 16-bit field which gives the rate of the base layer;
(iii) EL_i: a set of 16-bit fields which give the rate of the enhancement layer i, $i \in 1, \ldots, NL$;
(iv) k_i: a set of 8-bit fields conveying the rate of the Reed-Solomon code used for the protection of layer i, $i \in 0, \ldots, NL$.[1]

7. EXPERIMENTAL RESULTS

The performance of the SARC algorithm has been evaluated considering various sets of discriminative clustering variables using the NS2 (version 2.1b6), network simulator.

7.1. Analysis of fairness

The first set of experiments aimed at analyzing the fairness of the flows produced against conformant TCP flows. Fairness has been analyzed using the single bottleneck topology shown in Figure 4. In this topology, a number of sending nodes are connected to many receiving nodes via a common link with a bottleneck rate of 8 Mbps and a delay of 50 milliseconds. The video flows controlled by the SARC protocol are competing with 15 conformant TCP flows. Figure 5a depicts the respective throughput of one video

[1] Here we consider Reed-Solomon codes of rates k/n. The value of n is fixed at the beginning of the session and only the parameter k is adapted dynamically during the session. However, we could also easily consider adapting the parameter n, therefore the syntax of the SR packet would have to be extended accordingly.

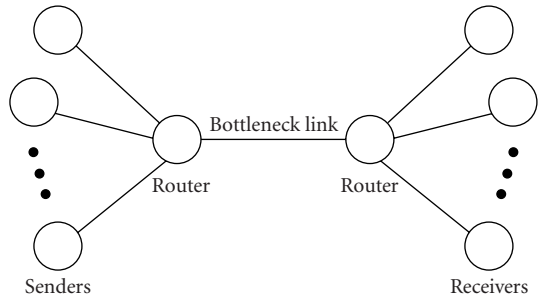


Figure 4: Simulation topology (bottleneck).

flow controlled with the goodput measure and of two out of the 15 TCP flows. Figure 5b depicts the throughputs obtained when using the TCP-compatible rate equation. As expected, the flow regulated with the goodput measure does not compete fairly with the TCP flows (cf. Figure 5a). In the presence of cross traffic at high rate, the EB decreases regularly to reach the lower bound $R_{\min}$ that has been set to 256 Kbps. The average throughput of the flow regulated with the TCP-compatible measure matches closely the average TCP throughput with a smoother rate (cf. Figure 5b).

7.2. Loss rate and PSNR performances

The second set of experiments aimed at measuring the PSNR and LR performances of the rate control mechanism, with two measures (goodput and TCP-compatible measures), with and without the presence of FEC. We have considered the multicast topology shown in Figure 6. The periodicity of the feedback rounds is set to be equal to the maximum RTT value of the set of receivers. The sequence used in the experiments, called "Brest,"[2] has a duration of 300 seconds (25 Hz, 6700 frames). The rate-distortion characteristics of the FGS source is depicted in Figure 7. The experiments depicted here are realized with the MoMuSys MPEG-4 version 2 video codec [9].

7.2.1. Testing scenario

Given the topology of the multicast tree, we have considered a source representation on three layers, each layer being transmitted to an IP multicast address. The BL is encoded at a constant bit rate of 256 Kbps. The overall rate (base layer plus two ELs) ranges from 256 Kbps up to 1 Mbps. At $t = 0$, each client subscribes to the three layers with respective initial rates of $R_{BL} = 256$ Kbps, $R_{EL1} = 100$ Kbps, and $R_{EL2} = 0$ Kbps. During the session, the video stream has to compete with point-to-point UDP cross traffic with a constant bit rate of 192 Kbps and with TCP flow. These competing flows contribute to a decrease of the links bottleneck. The activation of the cross traffic between clients represented by "squares" on Figure 6, in the time interval from 100 to 200 seconds, limits the bottleneck of the corresponding link (i.e., LAN 1's client) down to 320 Kbps. Similarly, competing TCP traffic is generated between clients denoted by "triangles" in the interval from 140 to 240 seconds leading to a bottleneck rate of the link (i.e., LAN 4's clients) down to 192 Kbps during the corresponding time interval.

The first test aimed at showing the benefits for the quality perceived by the receivers of an overall measure that would also take into account the source characteristics (and in particular the rate-distortion characteristics) versus a simple optimization of the overall goodput. Thus, we compare our results with the SAMM algorithm proposed in [3]. The corresponding mechanism is called SAMM-like in the sequel.

The SARC algorithm, relying on the rate-distortion optimization, has then been tested with, respectively, the goodput and the TCP-compatible measures in order to evidence the benefits of the TCP-compatible rate control in this layered multicast transmission system. In the sequel, these approaches are, respectively, called goodput-based source adaptive rate control (GB-SARC) and TCP-friendly source adaptive rate control (TCPF-SARC). The constant K is set to 4 in the experiments. In addition, in order to evaluate the impact of the FEC, we have considered the TCP-compatible bandwidth estimation both with and without FEC (TCPF-SARC+FEC) for protecting the BL. When FEC is not applied, the k_i parameter of each layer is set to n (i.e., 10 in the experiments).

7.2.2. Results

Figures 8 and 9 show the results obtained with the SAMM-like algorithm. It can be seen that the SAMM-like approach does not permit an efficient usage of the bandwidth. For example, the LAN 2's client (with a link with a bottleneck rate of 768 Kbps) has not received more than 300 Kbps on its link. Similar observations can be done with receivers of other LANs. Notice also that if the rate had not been lower bounded by an $R_{\min}$ value, the goodput of the different receivers would have converged to a very small value. In addition to the highly suboptimal usage of bandwidth, the approach suffers from a very unstable behavior in terms of subscriptions and unsubscriptions to multicast groups.

Figures 10, 11, and 12 show the rate variations of the different layers of the FGS source over the session, obtained, respectively, with the GB-SARC, TCPF-SARC, and TCPF-SARC+FEC methods. Figures 13, 14, and 15 depict the throughput estimated with these three methods versus the real measures of goodput, the LR, the number of layers received, and the PSNR values observed for two representative clients (i.e., LAN 2 with a bottleneck rate of 768 Kbps and LAN 4 with a bottleneck rate of 384 Kbps).

Figures 10 and 13, with the GB-SARC algorithm, show that the rate control that takes into account the PSNR (or rate-distortion) characteristics of the source leads to a better bandwidth utilization than the SAMM-like approach. In addition, the throughput estimated follows closely the bottleneck rates of the different links. Moreover, the number of irrelevant subscriptions and unsubscriptions to multicast

[2] Courtesy of Thomson Multimedia R&D France.

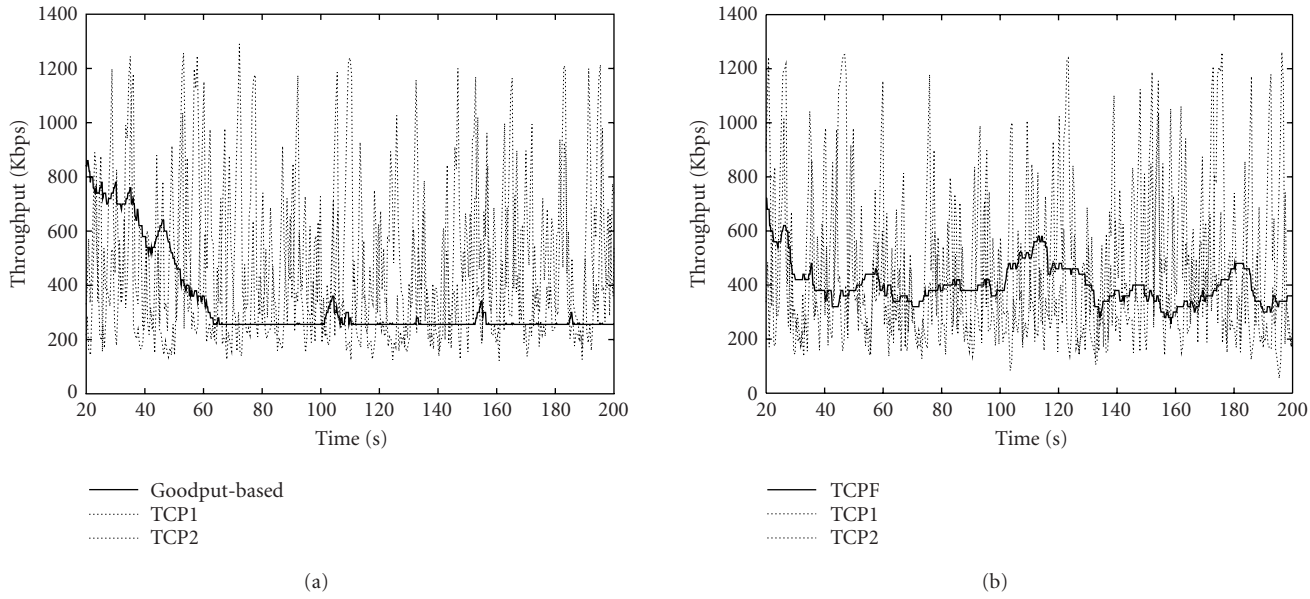


Figure 5: Respective throughputs of two TCP flows and of one rate-controlled flow with (a) a measure of goodput and (b) the TCP-compatible measure.

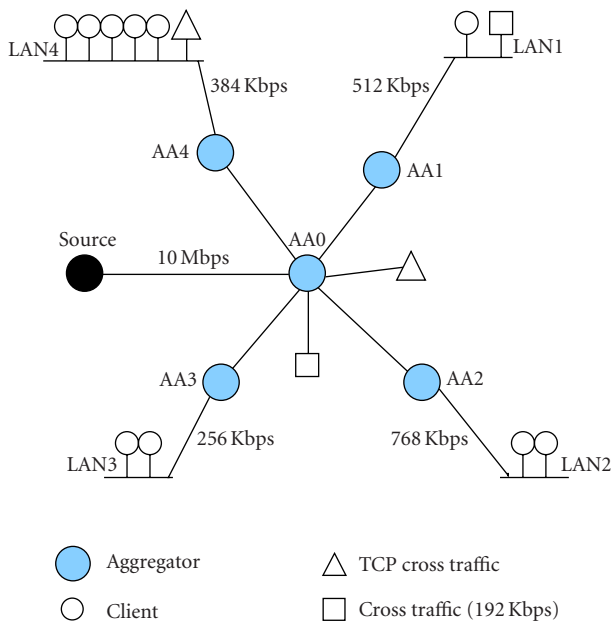


Figure 6: Simulated topology.

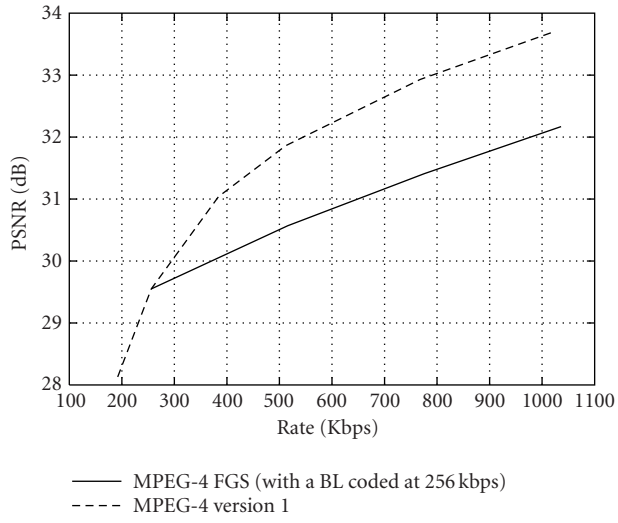


Figure 7: Rate-distortion model of the FGS video source.

groups is strongly reduced. However, the LRs observed remain high. For example, the LAN 4's client observe an average LR of 30% between 240 seconds and 300 seconds. This is due to the fact that during this time interval, the receiver of LAN 1 (bottleneck rate of 512 Kbps) has subscribed to the first enhancement layer (EL1), hence the rate of this layer is higher than the bottleneck rate of the LAN 4's clients. In this case, the GB-SARC algorithm does not permit a reliable bandwidth estimation for the LAN 4's clients. As expected, the quality of the received video suffers from the high LRs and the obtained PSNR values are relatively low. Finally, another important drawback is that during the corresponding period, the rate constraints given to the FGS video streaming server are very unstable (see Figure 10).

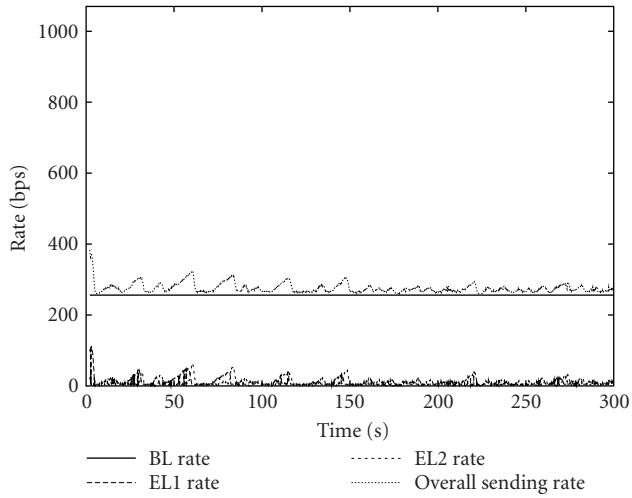
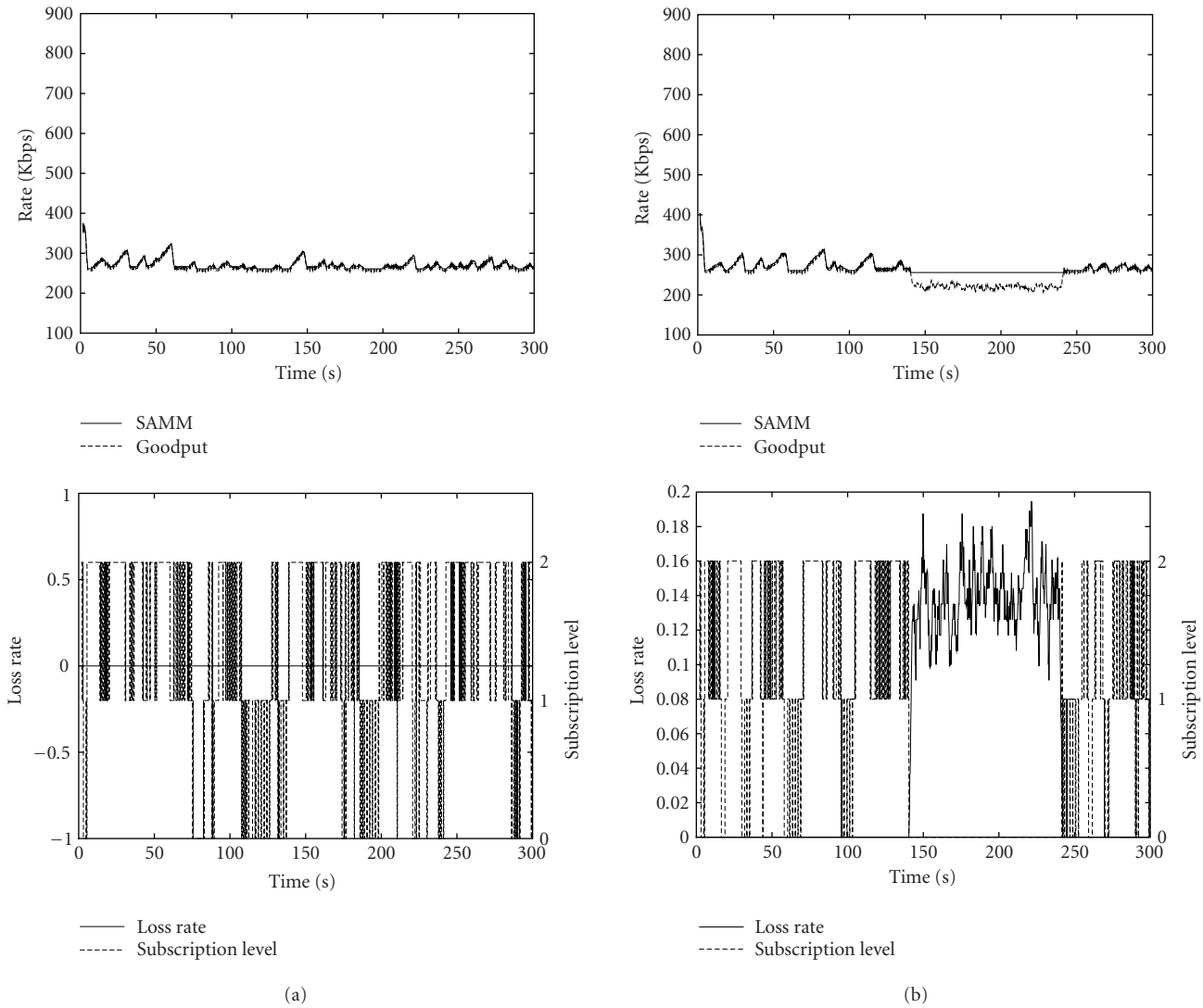


Figure 8: Rate variations for each layer of the FGS video source with the SAMM-like approach.



(a) (b)

Figure 9: SAMM-like throughput versus real goodput measure, LR, and subscription level obtained for (a) a LAN 2's client (link 768 Kbps) and (b) a LAN 4's client (link 384 Kbps).

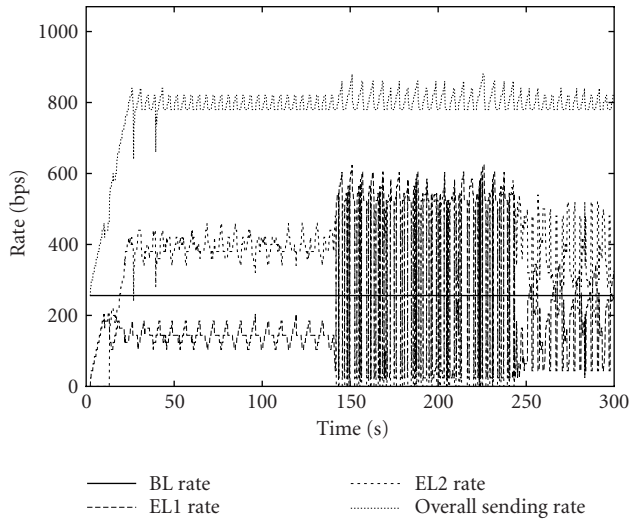


Figure 10: Rate variations for each layer of the FGS video source with the GB-SARC approach.

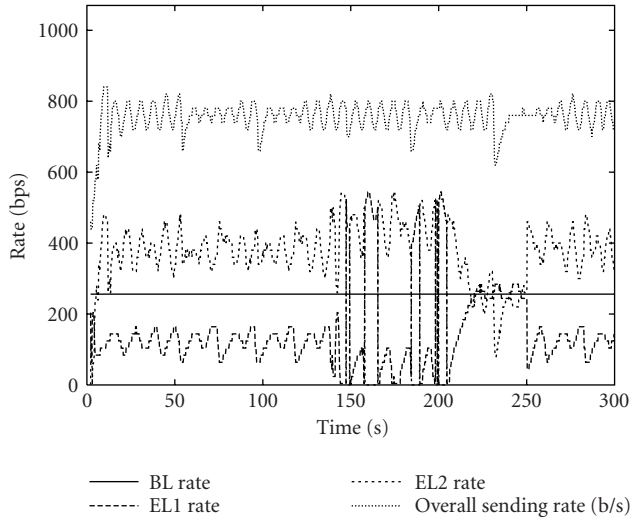


Figure 11: Rate variations for each layer of the FGS video source with the TCPF-SARC approach.

With the TCPF-SARC algorithm (cf. Figures 11 and 14), the sending rates of the different layers follows closely the variations of the bottleneck rates of the different links. This leads to stable sessions with low LRs and with a restricted number of irrelevant subscriptions and unsubscriptions to multicast groups. The comparison of the PSNR curves in Figure 14 reveals a gain of at least db for LAN 2 with respect to LAN 4. This evidences the interest of such multilayered rate control algorithm in a multicast heterogeneous environment. Notice that the peaks of instantaneous LRs observed result from a TCP-compatible prediction which occasionally exceeds the bottleneck rate. Also, in Figure 14b, the LR observed over the time interval from 140 to 240 seconds remains constant and relatively high. This comes from the fact that, in the presence of competing traffic, the bottleneck rate available for the video source is lower than the rate of the BL which in the particular case of an FGS source is maintained constant in average (e.g., 256 Kbps).

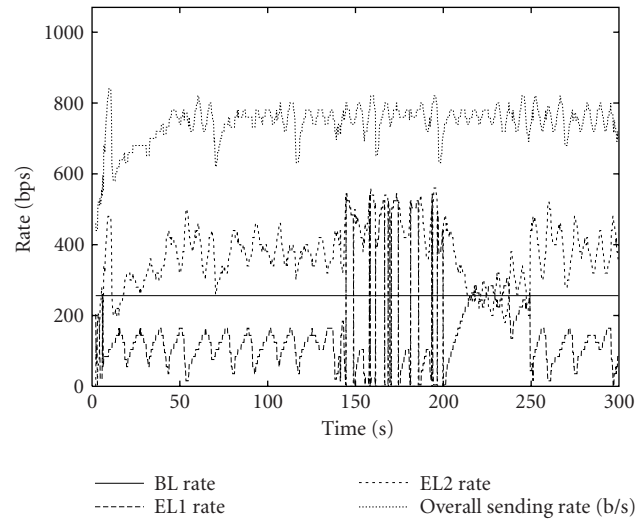


Figure 12: Rate variations for each layer of the FGS video source with the TCPF-SARC + FEC approach.

The FEC permits improving slightly the PSNR performances, especially for the receivers of LAN4 (cf. Figure 15b). It can be seen on Figure 12 that the usage of FEC however leads to a bit more unstable behavior, that is, to higher rate fluctuations of the different layers of the FGS source.

8. CONCLUSION

In this paper, we have presented a new multicast multilayered congestion control protocol called SARC. This algorithm relies on an FGS layered video transmission system in which the number of layers, their rate, as well as their level of protection are adapted dynamically in order to optimize the end-to-end QoS of a multimedia multicast session. A distributed clustering mechanism is used to classify receivers according to the packet LR and the bandwidth estimated on the path leading to them. Experimentation results show the ability of the mechanism to track fluctuation of the available bandwidth in the multicast tree, and at the same time the capacity to handle fluctuating LRs. We have shown also that using LR and TCP-compatible measures as discriminative variables in the clustering mechanism leads to higher overall PSNR (hence QoS) performances than using the LR and goodput measures.

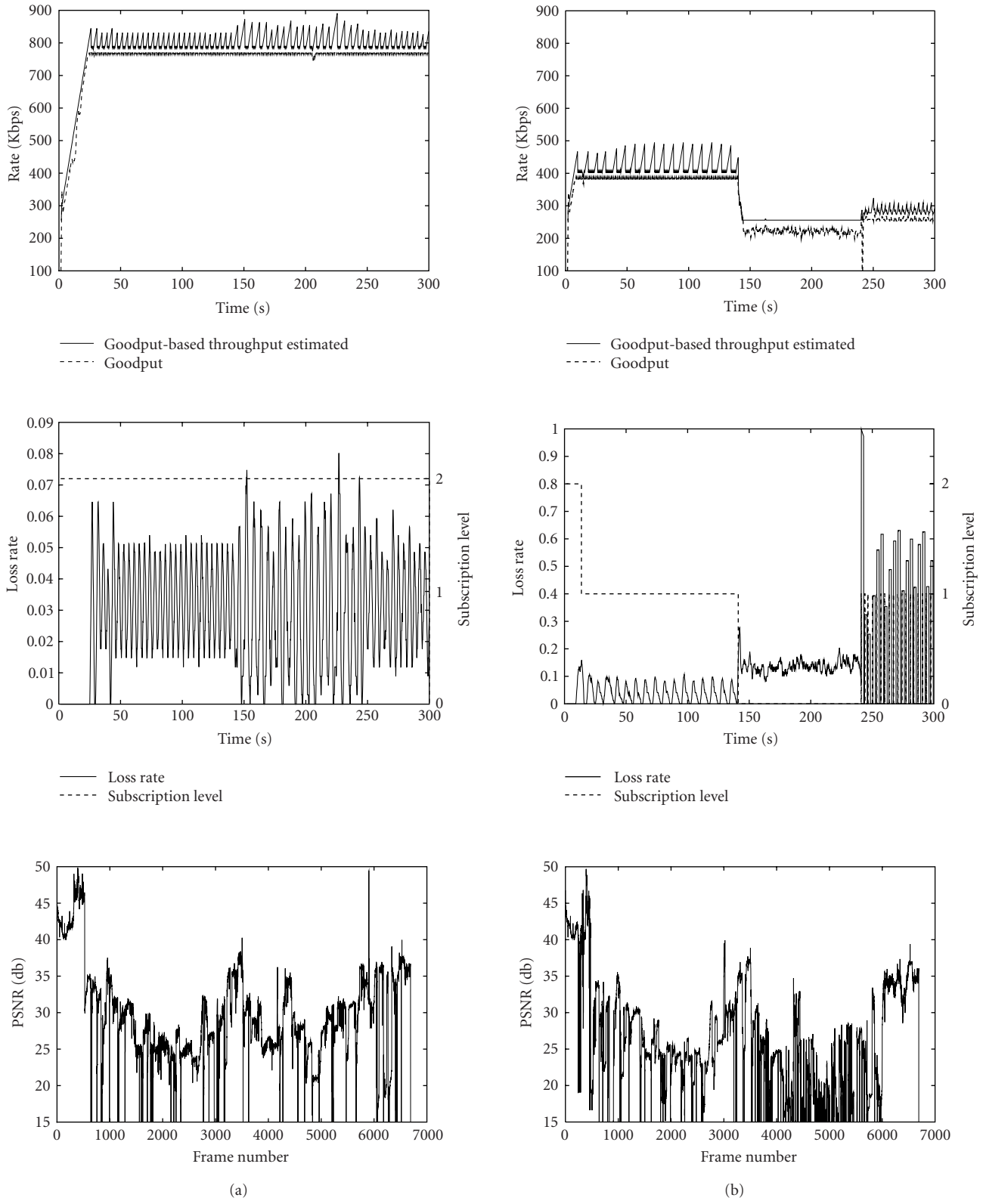


Figure 13: GB-SARC throughput versus real goodput measure, LR, subscription level, and PSNR obtained for (a) a LAN 2's client (link 768 Kbps) and (b) a LAN 4's client (link 384 Kbps).

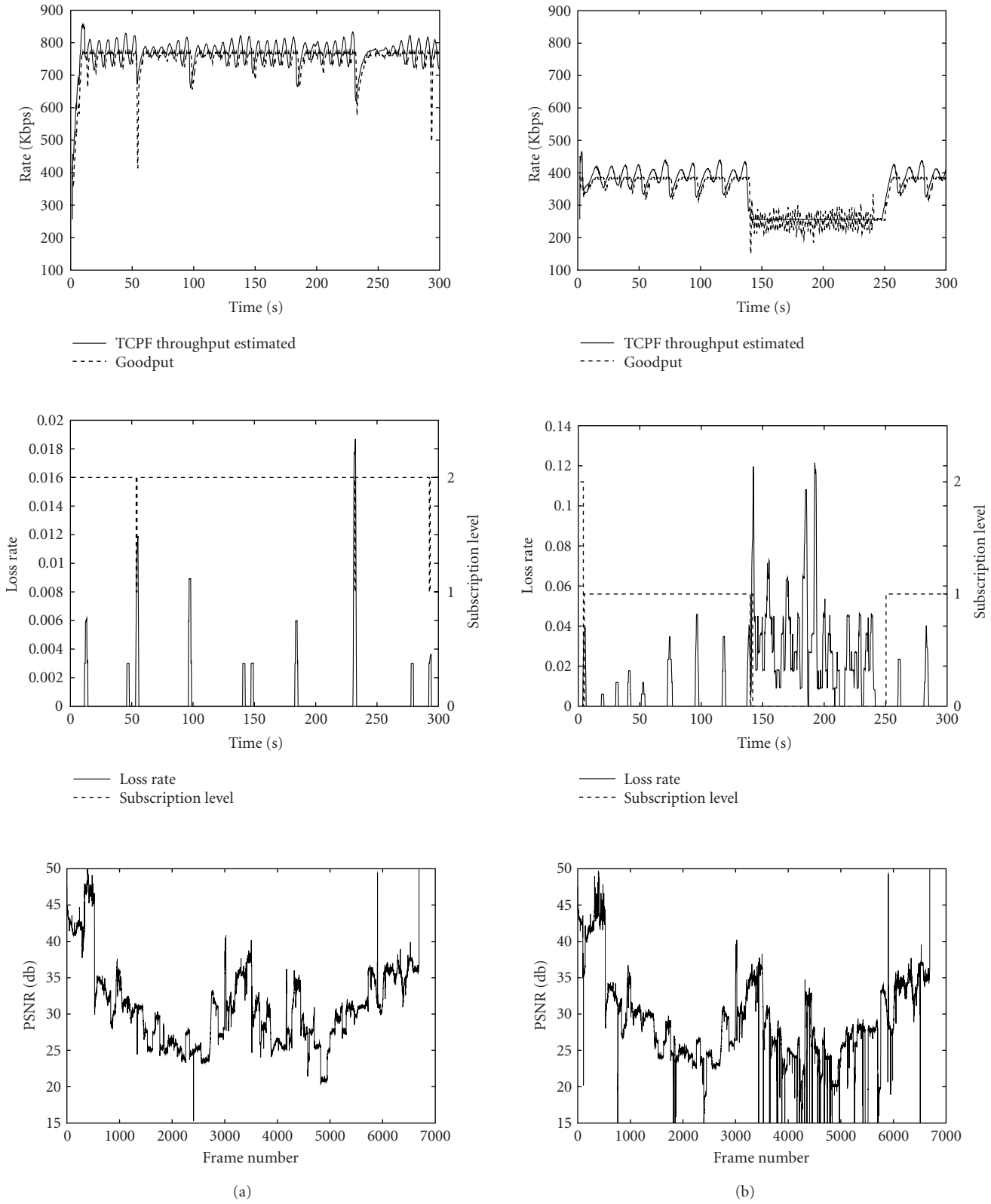


Figure 14: TCPF-SARC throughput versus real goodput measure, LR, subscription level, and PSNR obtained for (a) a LAN 2's client (link 768 Kbps) and (b) a LAN 4's client (link 384 Kbps).

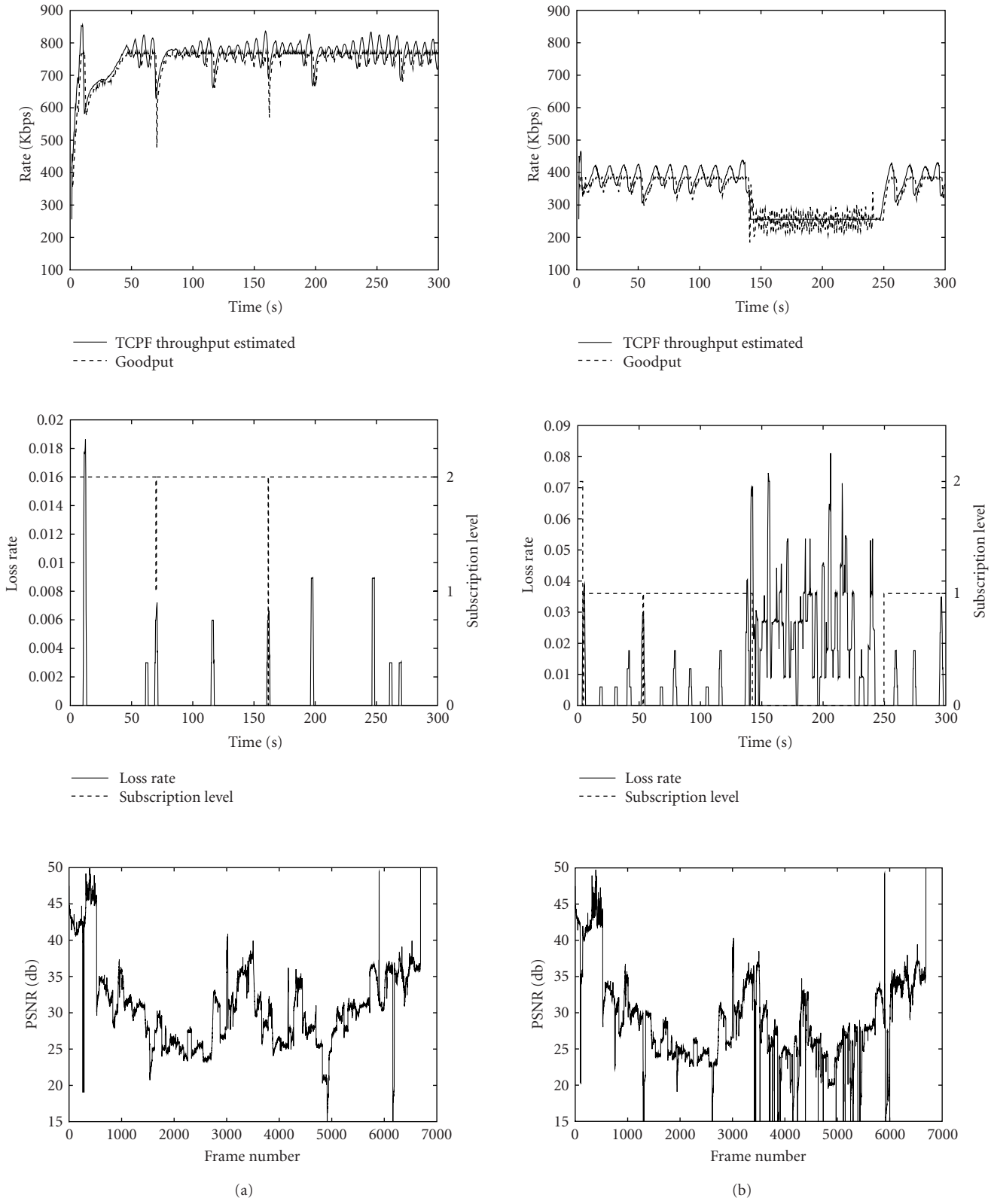


Figure 15: TCPF-SARC throughput with FEC versus real goodput measure, LR, subscription level, and PSNR obtained for (a) a LAN 2's client (link 768 Kbps) and (b) a LAN 4's client (link 384 Kbps).

REFERENCES

[1] S. McCanne, V. Jacobson, and M. Vetterli, "Receiver-driven layered multicast," in *Proc. Conference of the Special Interest Group on Data Communication (ACM SIGCOMM '96)*, pp. 117–130, Stanford, Calif, USA, August 1996.

[2] T. Turletti, S. Fosse-Parisis, and J. C. Bolot, "Experiments with a layered transmission scheme over the internet," Tech. Rep. RR-3296, INRIA, Sophia-Antipolis, 1997.

[3] B. J. Vickers, C. Albuquerque, and T. Suda, "Source adaptive multi-layered multicast algorithms for real-time video distribution," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, pp. 720–733, 2000.

[4] D. Sisalem and A. Wolisz, "MLDA: A TCP-friendly congestion control framework for heterogeneous multicast environments," Tech. Rep., GMD FOKUS, Berlin, Germany, 2000.

[5] Y. Wang and Q. F. Zhu, "Error control and concealment for video communication: A review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, 1998.

[6] J. C. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-based error control for internet telephony," in *Proc. Conference on Computer Communications (IEEE Infocom '99)*, pp. 1453–1460, NY, USA, March 1999.

[7] K. Salamatian, "Joint source-channel coding applied to multimedia transmission over lossy packet network," in *Proc. Packet Video Workshop (PV '99)*, NY, USA, April 1999.

[8] H. Radha and Y. Chen, "Fine granular scalable video for packet networks," in *Proc. Packet Video Workshop (PV '99)*, Columbia University, NY, USA, April 1999.

[9] Mobile Multimedia Systems (MoMuSys) Software, "MPEG-4 video verification model 4.1", December 2000.

[10] J. C. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the internet," in *Proc. Conference of the Special Interest Group on Data Communication (ACM SIGCOMM '94)*, pp. 58–67, London, UK, September 1994.

[11] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 6, pp. 982–1001, 1997.

[12] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," in *Proc. Conference on Computer Communications (IEEE Infocom '98)*, pp. 996–1003, San Francisco, Calif, USA, March 1998.

[13] D. Sisalem and A. Wolisz, "MLDA: A TCP-friendly congestion control framework for heterogeneous multicast environments," in *Proc. International Workshop on Quality of Service (IWQoS '00)*, Pittsburgh, Pa, USA, June 2000.

[14] X. Hénocq, F. Le Léannec, and C. Guillemot, "Joint source and channel rate control in multicast layered video transmission," in *Proc. SPIE International Conference on Visual Communication and Image Processing (VCIP '00)*, pp. 296–307, Perth, Australia, June 2000.

[15] A. Legout and E. W. Biersack, "Pathological behaviors for RLM and RLC," in *Proceedings of International Conference on Network and Operating System Support for Digital Audio and Video (NOSSDAV '00)*, pp. 164–172, Chapel Hill, NC, USA, June 2000.

[16] A. Legout and E. W. Biersack, "PLM: Fast convergence for cumulative layered multicast transmission schemes," in *Proc. ACM (SIGMETRICS '00)*, pp. 13–22, Santa Clara, Calif, USA, 2000.

[17] S. Bhattacharya, D. Towsley, and J. Kurose, "The loss path multiplicity problem in multicast congestion control," in *Proc. Conference on Computer Communications (IEEE Infocom '99)*, vol. 2, pp. 856–863, NY, USA, March 1999.

[18] J. Widmer and M. Handley, "Extending equation-based congestion control to multicast applications," in *Proc. Conference of the Special Interest Group on Data Communication (ACM SIGCOMM '01)*, pp. 275–286, San Diego, Calif, USA, August 2001.

[19] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proc. Conference of the Special Interest Group on Data Communication (ACM SIGCOMM '00)*, pp. 43–56, Stockholm, Sweden, August 2000.

[20] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shave, "FLID-DL: Congestion control for layered multicast," in *Proc. Second International Workshop on Networked Group Communication (NGC '00)*, pp. 71–81, Palo Alto, Calif, USA, November 2000.

[21] M. Luby and V. k. Goyal, "Wave and equation based rate control building block," Internet Engineering Task Force, Internet Draft draft-ietf-rmt-bb-webrc-04, June 2002.

[22] Q. Guo, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "A sender-adaptive and receiver-driven layered multicast scheme for video over internet," in *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS '01)*, Sydney, Australia, May 2001.

[23] K. Salamatian and T. Turletti, "Classification of receivers in large multicast groups using distributed clustering," in *Proc. Packet Video Workshop (PV '01)*, Taejon, Korea, May 2001.

[24] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP thoughput: a simple model and its empirical validation," in *Proc. Conference of the Special Interest Group on Data Communication (ACM SIGCOMM '98)*, pp. 303–314, University of British Columbia, Vancouver, Canada, August 1998.

[25] J. Viéron and C. Guillemot, "Real-time constrained TCP-compatible rate control for video over the internet," to appear in IEEE Transactions on Multimedia.

[26] M. Yajnik, J. Kurose, and D. Towsley, "Packet loss correlation in the MBone multicast network," in *Proc. IEEE Global Internet Conference*, London, UK, November 1996.

[27] B. N. Levine, S. Paul, and J. J. Garcia-Luna-Aceves, "Organizing multicast receivers deterministically by packet-loss correlation," in *Proc. 6th ACM International Conference on Multimedia (ACM Multimedia 98)*, Bristol, UK, September 1998.

[28] S. Paul, K. K. Sabnani, J. C. Lin, and S. Bhattacharya, "Reliable multicast transport protocol (RMTP)," *IEEE Journal On Selected Areas in Communications*, vol. 15, no. 3, pp. 407–421, 1997.

[29] R. El-Marakby and D. Hutchison, "Scalability improvement of the real-time control protocol (RTCP) leading to management facilities in the internet," in *Proc. 3rd IEEE Symposium on Computers and Communications (ISCC '98)*, pp. 125–129, Athens, Greece, June 1998.

[30] K. L. Calvert, J. Griffioen, B. Mullins, A. Sehgal, and S. Wen, "Concast: Design and implementation of an active network service," *IEEE Journal on Selected Area in Communications (JSAC)*, vol. 19, no. 3, pp. 720–733, 2001.

[31] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantiser design," *IEEE Transactions on Communications*, vol. 28, pp. 84–95, January 1980.

[32] F. J. Mac Williams and N. J. A. Sloane, *The Theory of Error Correcting Codes*, North Holland, Amsterdam, 1977.

[33] D. Koo, *Elements of Optimization*, Springer-Verlag, NY, USA, 1977.

[34] D. Tan and A. Zakhor, "Video multicast using layered FEC and scalable compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 3, pp. 373–387, 2001.

Jérôme Viéron received his M.S. degree in computer science from the University of Rennes, France, in 1999. From 1999 to 2003, he was pursuing his Ph.D. works at INRIA. He received his Ph.D. degree in computer science from the University of Rennes, France, in 2003. Currently he is with the Corporate Research Center of Thomson Multimedia R&D in Rennes, France. He works in the Multimedia Streaming & Storage Lab. His research interests are new generation scalable video compression for TV, HDTV, and digital cinema.

Thierry Turletti received his M.S. and Ph.D. degrees in computer science, both from the University of Nice Sophia-Antipolis, France, in 1990 and 1995, respectively. He has done his Ph.D. studies in the RODEO group at INRIA Sophia Antipolis. During 1995–1996, he was a Postdoctoral Fellow in the Telemedia, Networks and Systems Group at the MIT Laboratory for Computer Science (LCS), Massachusetts Institute of Technology (MIT). He is currently a Research Scientist at the Planète group at INRIA Sophia Antipolis. His research interests include multimedia applications, congestion control, and wireless networking. Dr. Turletti currently serves on the editorial board of Wireless Communications and Mobile Computing.

Kavé Salamatian is an Associate Professor at Paris VI University in France and conducts his researches at LIP6. His main areas of research are networking information theory and Internet measurement and modelling. He is actually the coordinator of a large research effort in Internet measurement and modelling in France. He has graduated in 1998 from Paris-SUD Orsay University with a Ph.D. degree in computer science. He worked during his Ph.D. on joint source-channel coding applied to multimedia transmission over Internet. Dr. Salamatian also has an M.S. in theoretical computer science from Paris XI University (1996) and an M.S. in communication engineering from Isfahan University of Technology (1995).

Christine Guillemot is currently "Directeur de Recherche" at INRIA, in charge of a research group dealing with image modelling, processing, and video communication. She holds a Ph.D. degree from Ecole Nationale Supérieure des Telecommunications (ENST), Paris. From 1985 to October 1997, she has been with CNET France Telecom where she has been involved in various projects in the domain of coding for TV, HDTV, and multimedia applications. From January 1990 to mid 1991, she has worked at Bellcore, NJ, USA, as a Visiting Scientist. Her research interests are signal and image processing, video coding, and joint source and channel coding for video transmission over the Internet and over wireless networks. She currently serves as Associated Editor for IEEE Transactions on Image Processing.